convolutional neural networks, swift and iOS 12

by brett koonce

september 19th, 2018

overview

- **background, what to study**
- **neural networks, convolutions**
- **mobile image recognition**
- **next steps**

enter the temple

training

- -1) python, roulette
- 0) calc/linear algebra basics
- 1) coursera, keras + tensorflow
- 2) fast.ai 2018 sequence, pytorch
- 3) read, practice, get into real world

platform

- 0) math + cpu + tools
- 1) basic vm's (coursera, paperspace)
- 2) cloud software (aws, gcp)
- 3) edge: mobile devices/embedded
- 4) custom hardware (tpu, volta, asic)

machine learning

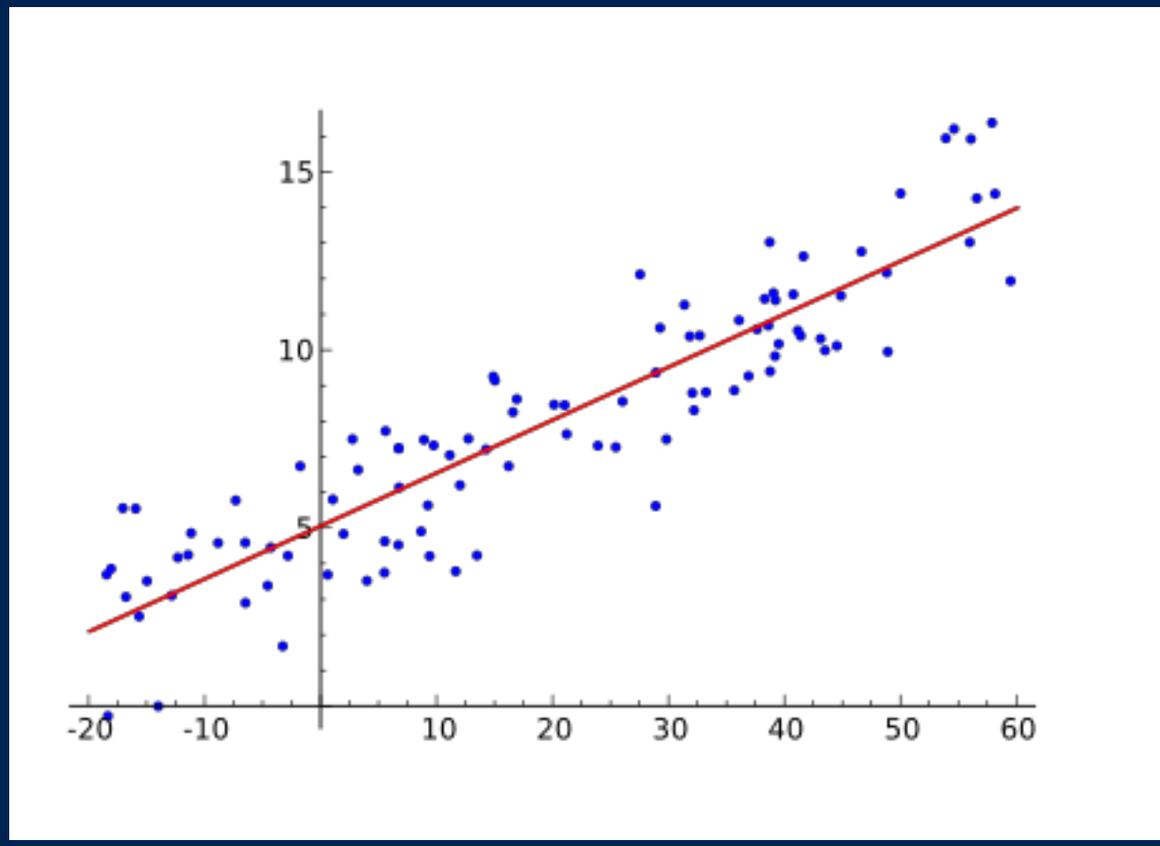
- **supervised**
- **unsupervised**
- **reinforcement**

ml concepts

- an input (numbers, image, audio, video)
- known data (supervised learning)
- combine to produce function/black box
- train model, use on unknown data
- goals: quality, size, complexity

linear regression

- points —> slope == linear regression
- think of adding all the points you have as one epoch



neural network

- 1 layer, 1 node nn == perceptron
- 2 layer, x nodes nn == neural network
- feed forward neural network
- tensorflow.js mnist layers example

Neural Networks

- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

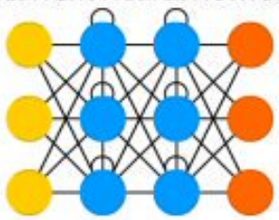
Perceptron (P)

Feed Forward (FF)

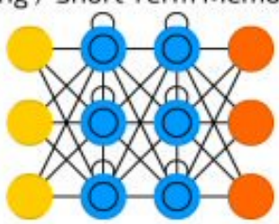
Radial Basis Network (RBF)

Deep Feed Forward (DFF)

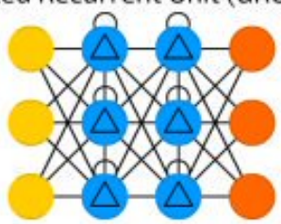
Recurrent Neural Network (RNN)



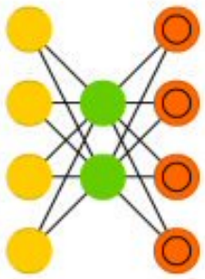
Long / Short Term Memory (LSTM)



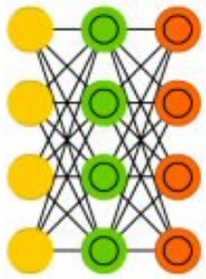
Gated Recurrent Unit (GRU)



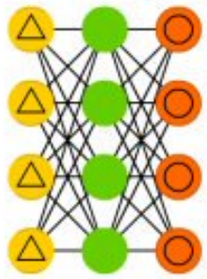
Auto Encoder (AE)



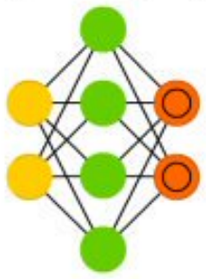
Variational AE (VAE)



Denoising AE (DAE)



Sparse AE (SAE)

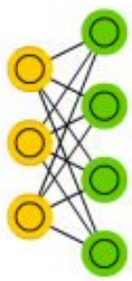


Markov Chain (MC)

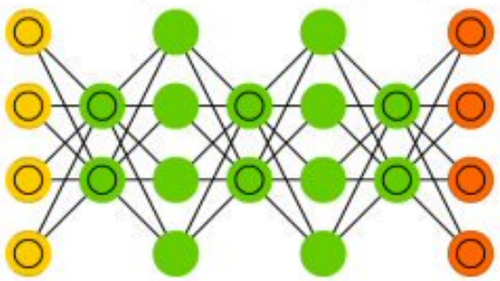
Hopfield Network (HN)

Boltzmann Machine (BM)

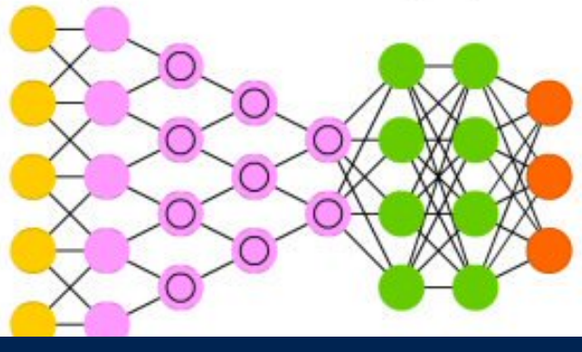
Restricted BM (RBM)



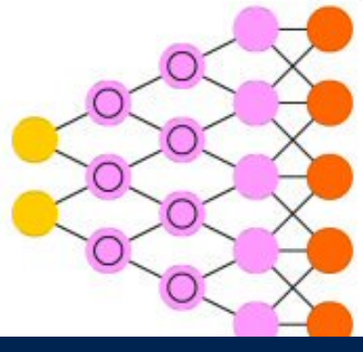
Deep Belief Network (DBN)



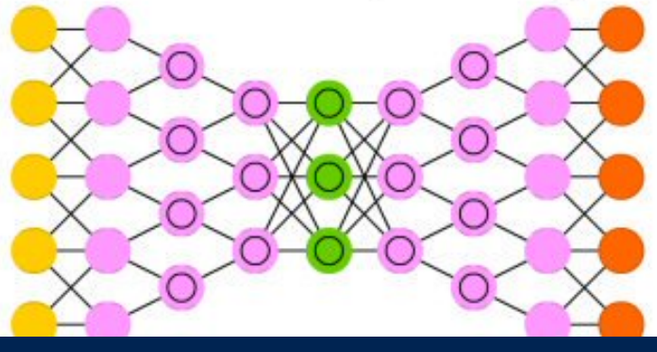
Deep Convolutional Network (DCN)



Deconvolutional Network (DN)



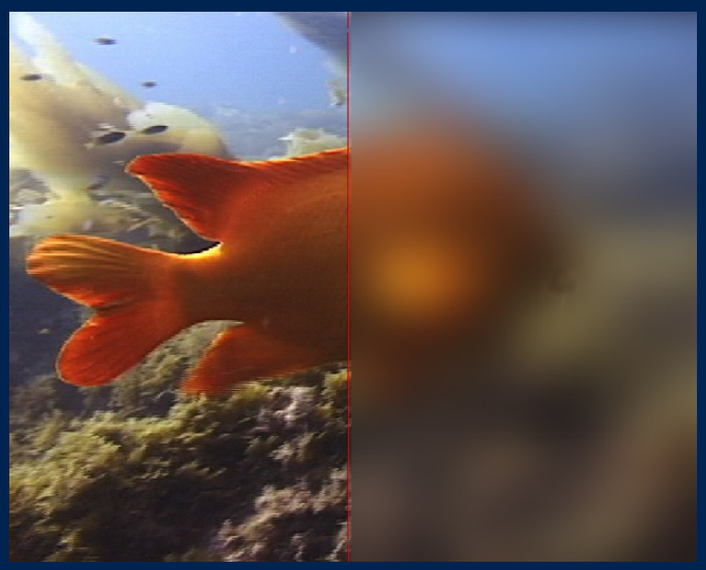
Deep Convolutional Inverse Graphics Network (DCIGN)



convolutions

- **convolution == matrix math == a[x] + b**
- **transform[image] + offset == i1 —> i2**

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16



-1	0	1
-2	0	2
-1	0	1

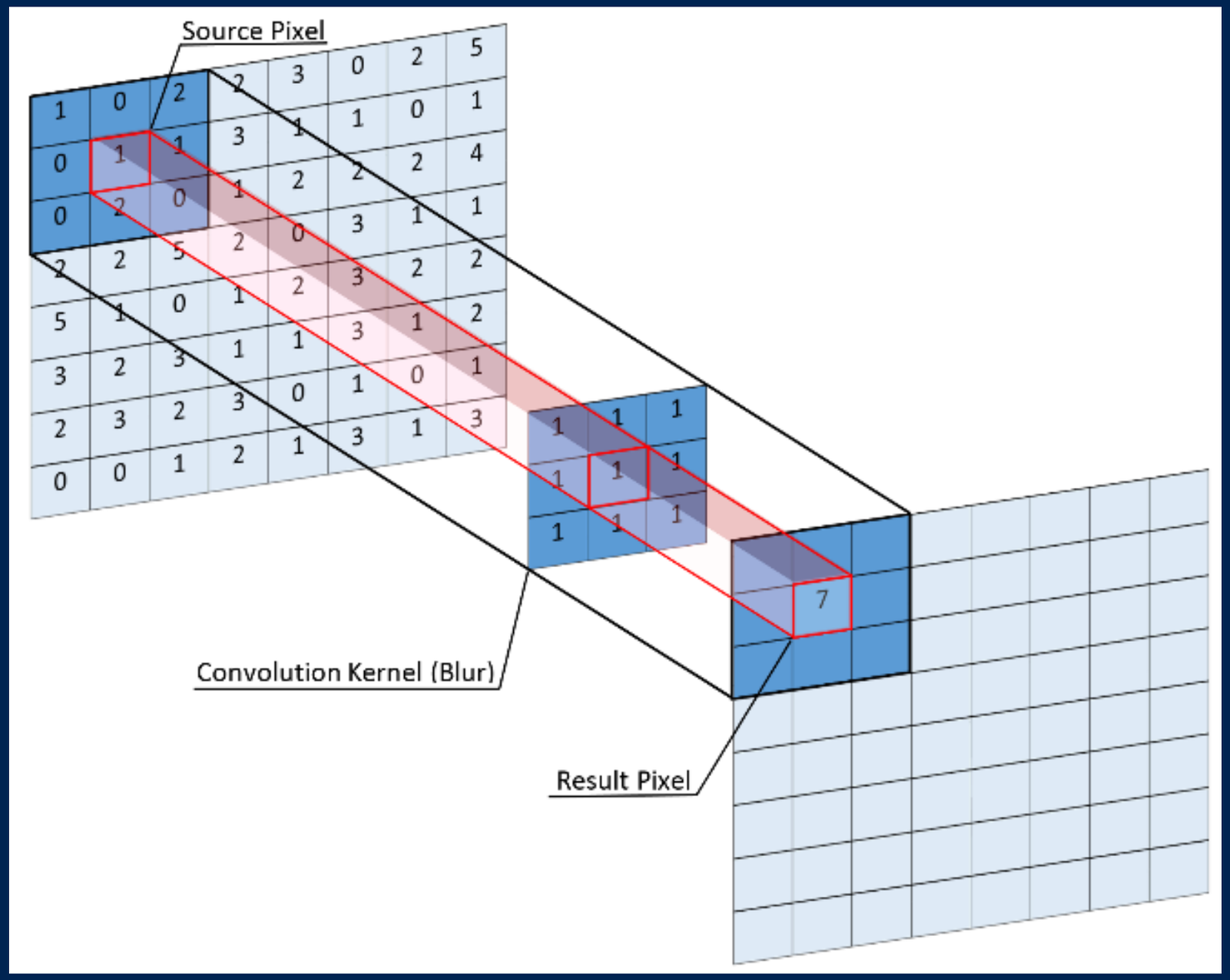
Horizontal

-1	-2	-1
0	0	0
-1	-2	-1

Vertical

convolutions v2

- [LxM] => [MxN]
- data A —> data B

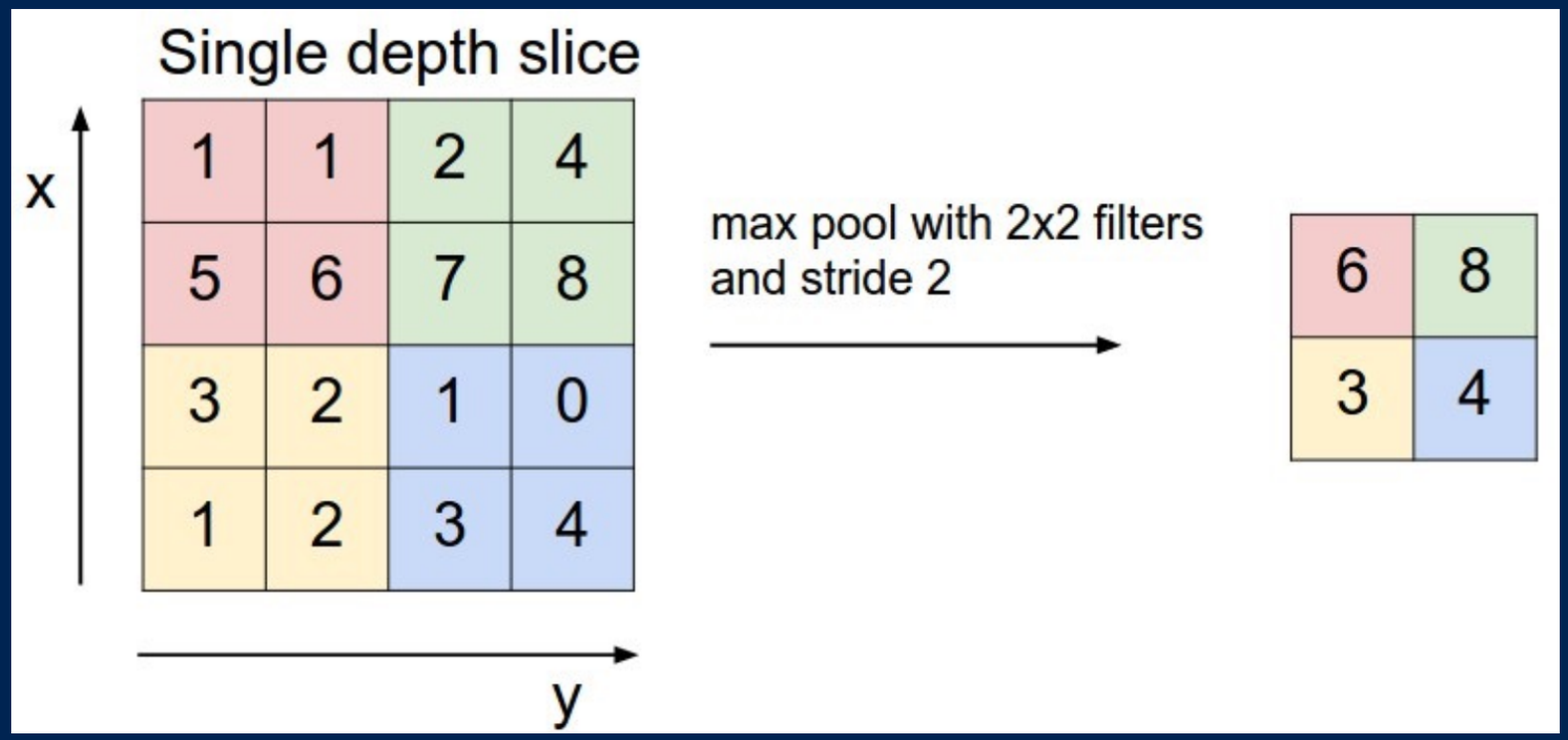


conv: 3x3 striding

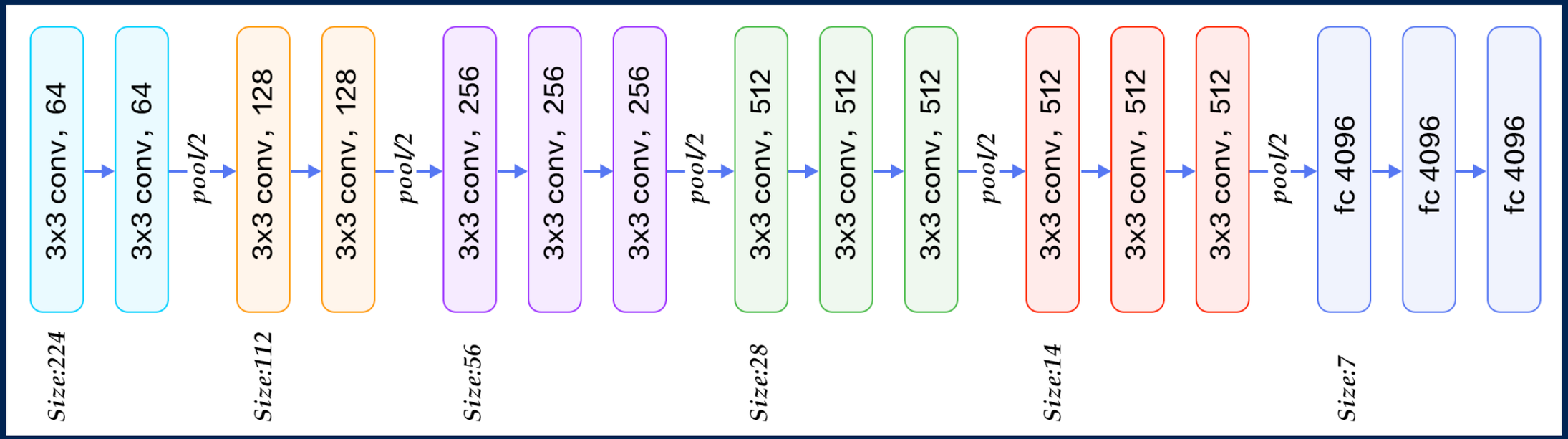
- break input image into smaller chunks
- cs231n.github.io
- deepfish

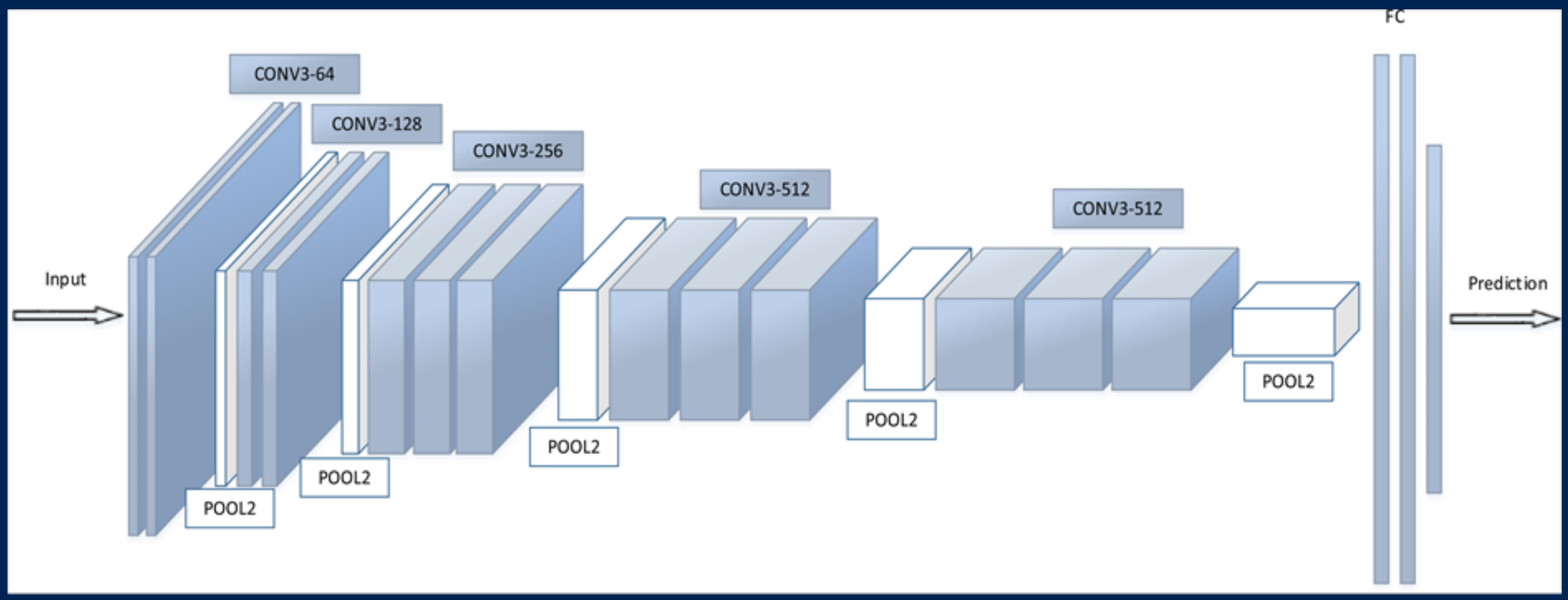
conv: 2x2 maxpool

- striding produces a lot of samples
- next: downsample the data!



combined: vggnet





resnet

- residual networks
- skip connections
- deeper training
- average pooling
- turicreate demo

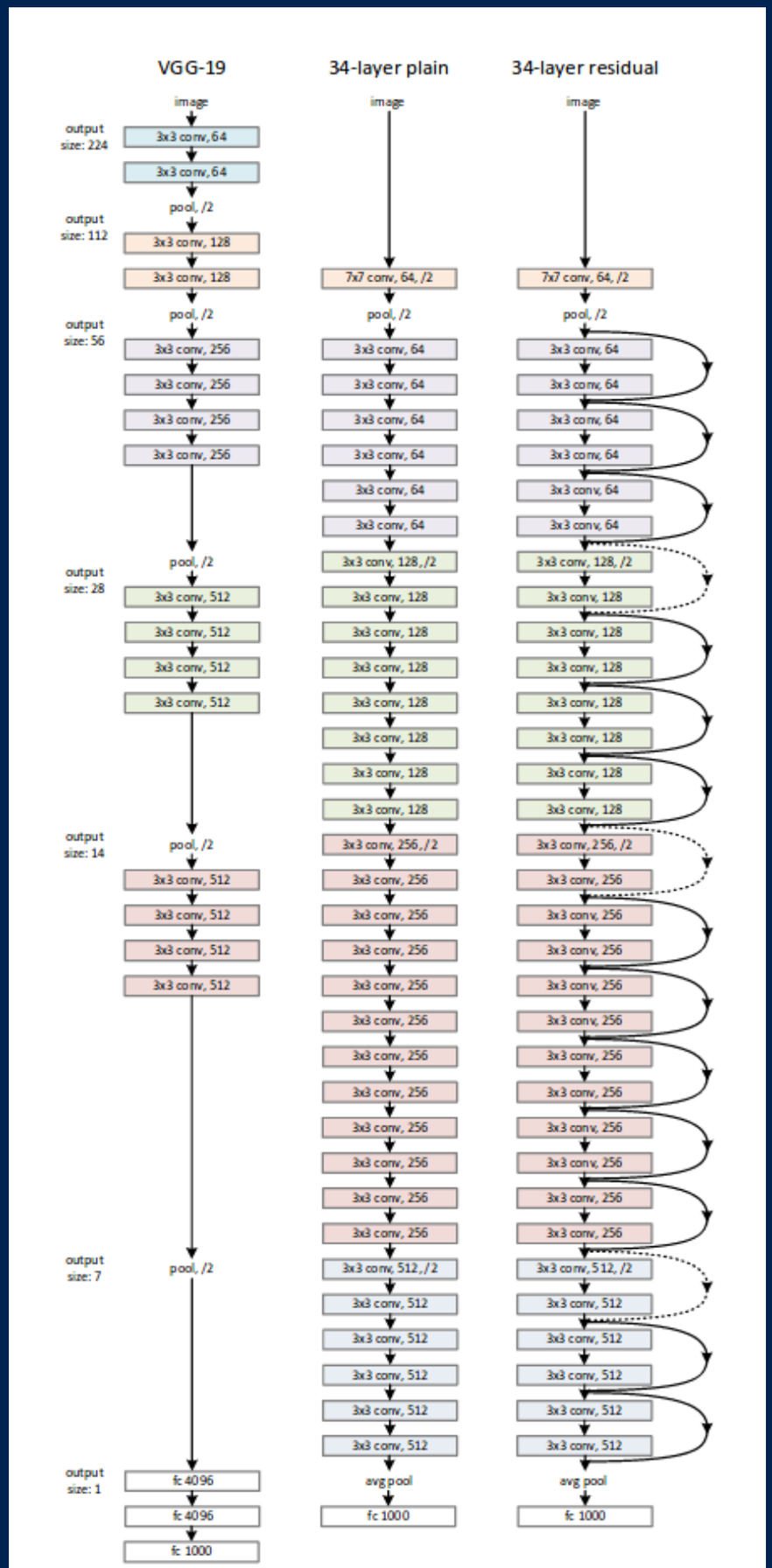


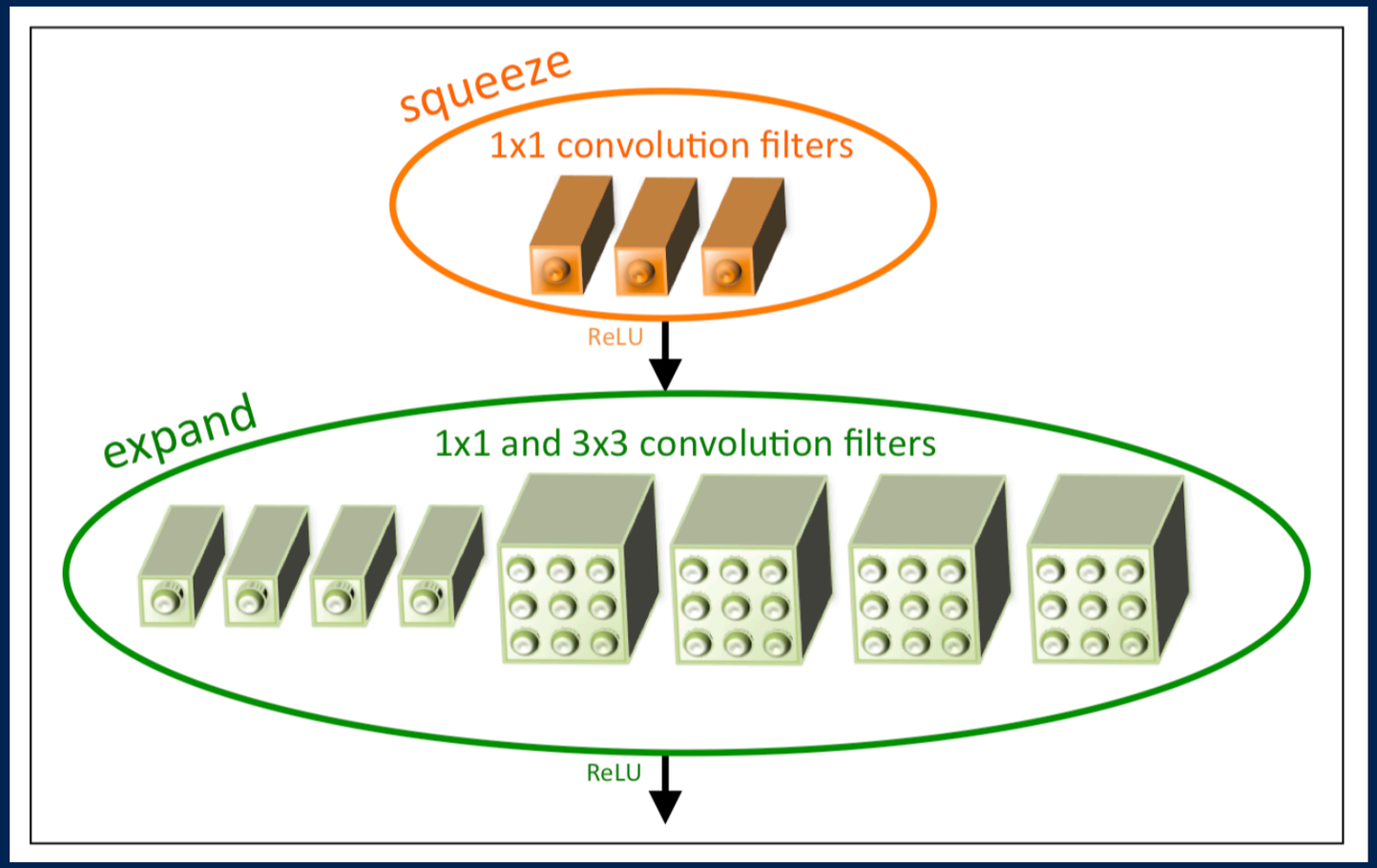
image recognition

image recognition

- 2017 talk: basic nn, vgg, inception, resnet, mobilenets, (yolo)
- general: densenet, darknet, unet
- mobile: squeezenet, mobilenets v1, shufflenet, senet, mn v2, nasnet mobile

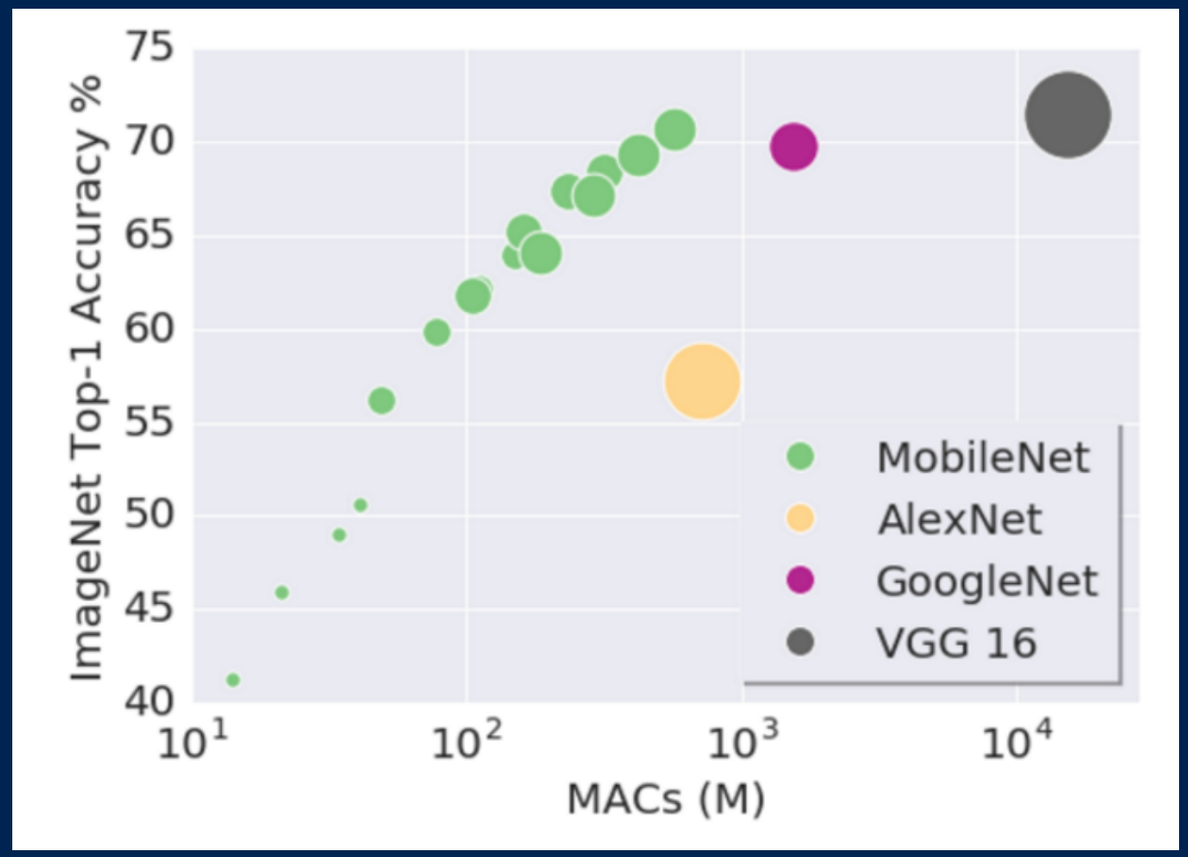
squeezenet

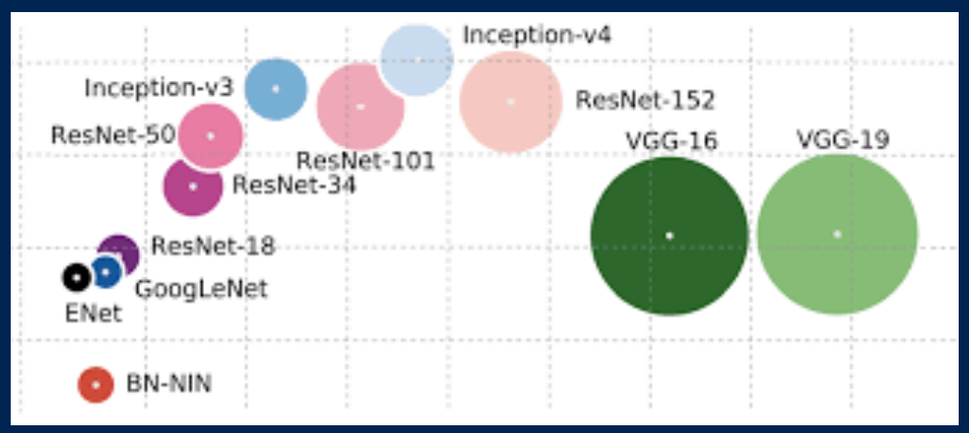
- 2016
- fire modules
- turicreate
- 5MB —> 80% accuracy



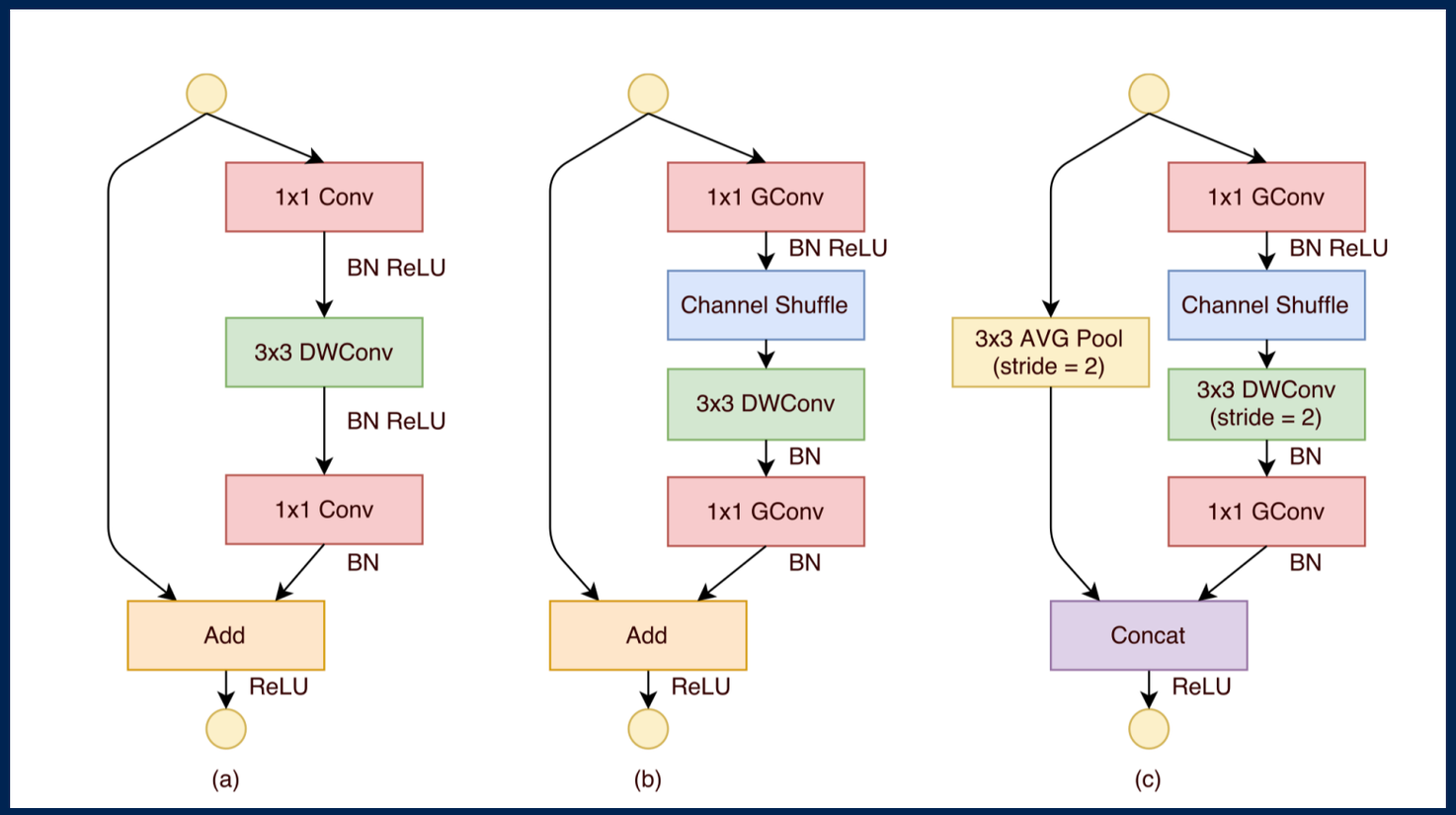
mobilenets

- depthwise separable convolutions
- april 17, paper
- tf-lite demo
- createml





shufflenet



- 2017, pointwise depth convolutions + channel shuffle
- https://medium.com/syncedreview/shufflenet-an-extremely-efficient-convolutional-neural-network-for-mobile-devices-72c6f5b01651

senet

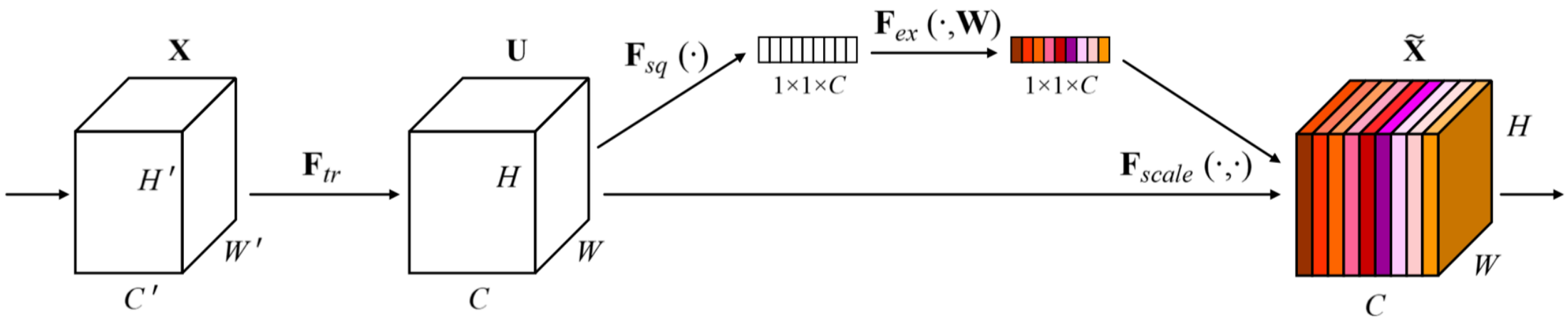


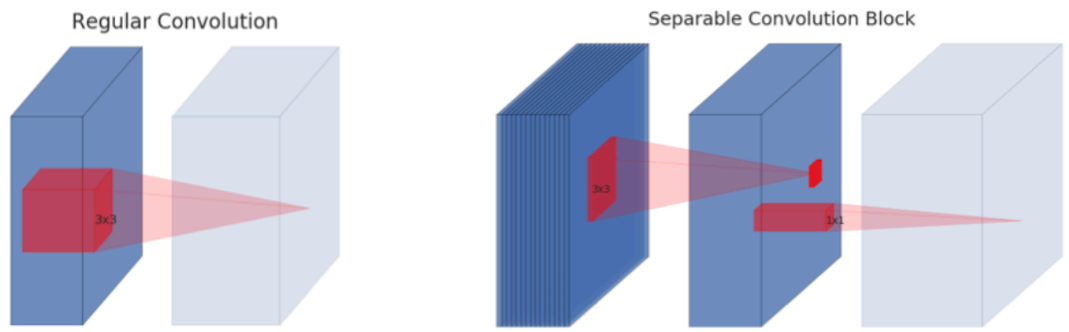
Figure 1: A Squeeze-and-Excitation block.

- **squeeze + excitation networks (2018)**
- **https://www.robots.ox.ac.uk/~vgg/publications/2018/Hu18/presentation.pdf**

mobilenets v2

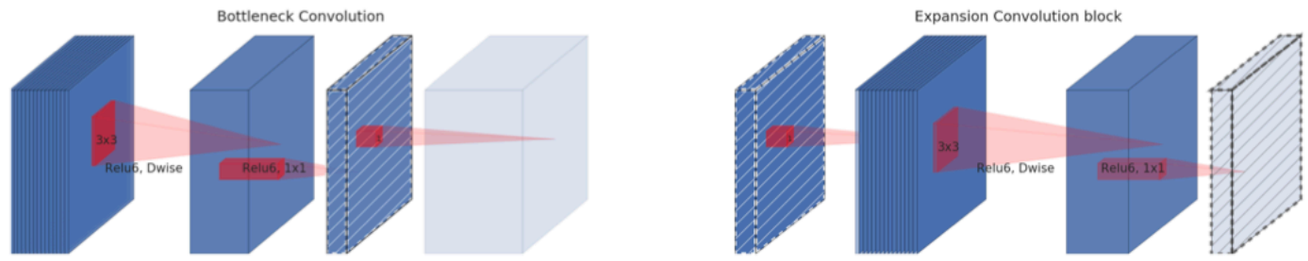
(a) Regular

(b) Separable



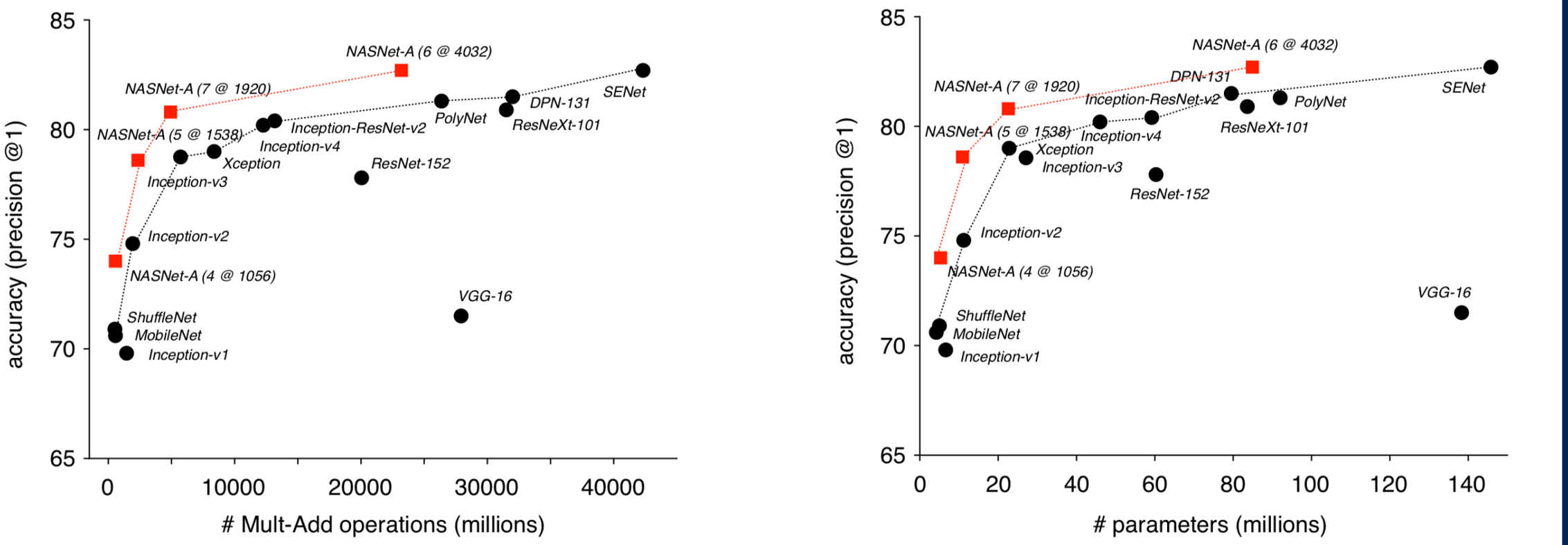
(c) Separable with linear bottleneck

(d) Bottleneck with expansion layer



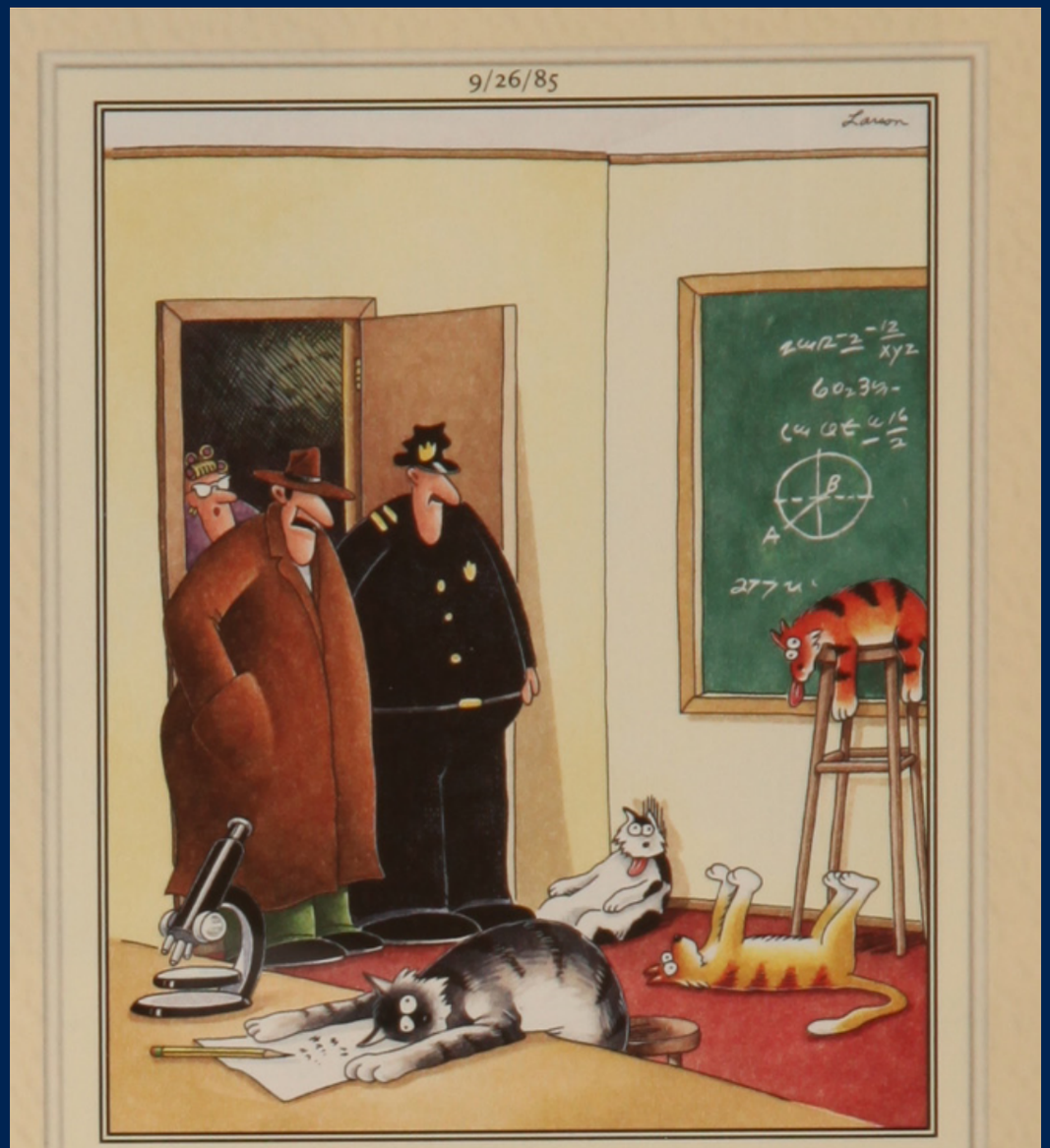
- **2018, tensorflow lite**
- **http://machinethink.net/blog/mobilenet-v2/**

nasnet



- **nasnet, nasnet mobile (2017)**
- **automl, amoebanet (2018)**

look to the wind



"Notice all the computations, theoretical scribblings, and lab equipment, Norm. ... Yes, curiosity killed these cats."

pytorch + caffe

- upcoming 1.0 release (october 2018)
- pytorch —> onnx —> caffe
- mxnet, cntk, coreml interop

tensorflow + swift

- eager execution —> tf2
- swift type safety + llvm + lattner + google —> cool stuff
- tpu demo, tf dev conference
- scale by the bay talk: nov 17th

next steps

- **recurrent neural networks**
- **gan/rl —> ?? —> $$**
- **language models**
- **...**